REV 2 — 2026
LARS BLOMGAARD
www.defencia.dk

Steps for Analysing Acquired Data

Digital Forensics & Incident Response — Evidence Analysis Procedure

PRINCIPLE	This is not an exhaustive list — cases vary significantly. Apply these steps based on the hypothesis and scenario. Always document every action taken. If it is not written down, it did not happen.

1 — Case Preparation

1.1 Evidence Metadata from Acquisition

Record all of the following before beginning analysis. This forms the foundation of the chain of custody.

Case description	What is this case about? Short, factual description. Incident type.
Analyst responsible	Full name, role, contact information.
Start date & time (UTC)	When did analysis begin? Include UTC offset of analyst and target system.
NTP / Timezone	CRITICAL: Document the NTP server and timezone of the source system. All timestamps must be normalised to UTC.
Number of evidence items	How many items were analysed in this case?
Tools & versions	All tools used: Autopsy, EnCase, Magnet AXIOM, X-Ways, Volatility, KAPE, etc. Include exact version numbers.
Additional information	Any supplementary notes, access restrictions, or special circumstances.
Hypothesis	What is the working hypothesis? Provide a quick overview of all hypotheses surrounding the case. These drive the analysis scope.

1.2 Constraints & Limitations

Document any boundary that affects scope or completeness. Be precise and concise.

- Time limitation — legal deadline, business continuity, on-call window
- Specific search scope — hypothesis-driven or targeted artefact search only
- IOC / YARA / HASH targeted search — note the search method and vendor file identification used
- SSD limitations — TRIM may have deleted files during or before collection: document and note impact on findings
- Encrypted volumes — BitLocker, LUKS, FileVault: was key obtained? Was decryption possible?
- Cloud or remote evidence — limitations on acquisition completeness
- Legal / scope constraints — GDPR, PII handling, court orders, jurisdiction
- Chain of custody gaps — document any breaks or uncertainties

2 — Analysis on Evidence

2.1 Malware Indicators

- Known malware files identified by AV or threat intel vendors?
  - IOCs tested against evidence — list all IOCs used and results
- Common malware artefacts checked:
  - New / unexpected files spawned in system directories
  - Contact to known malicious domains or IPs
  - Unusual activity in user folders (executables, .lnk files, etc.)
  - Timeline activity correlating with suspicious events
  - Files downloaded from the internet (Zone.Identifier ADS streams)
  - Scheduled tasks — new or modified (schtasks, Task Scheduler XML)
- YARA rules applied — document rule names, sources, and hits

2.2 Scripting, Living-off-the-Land & Privilege Escalation

- Signs of scripting activity
  - PowerShell: ScriptBlock logging (Event ID 4104), ConsoleHost_history.txt, PSReadLine
  - WMI subscriptions: check for malicious WMI event consumers
  - DLL sideloading, DLL search order hijacking
  - Python, VBScript, AutoIt, or other interpreted languages
  - LOLBins (Living-off-the-Land Binaries): certutil, mshta, regsvr32, rundll32, wscript, cscript, bitsadmin
- Privilege escalation
  - UAC bypass techniques identified?
  - Token impersonation or access token manipulation
  - Software installations performed with elevated / privileged accounts
  - Unexpected accounts added to local Administrators or Domain Admins

2.3 USB & External Devices

- How many external devices were connected to the system?
- Device details: make, model, serial number (from registry: USBSTOR, USB hive keys)
- First connected / last connected timestamps
- Drive letter assigned — volume name
- Files transferred to/from device (correlation with MFT / LNK / shellbags / jumplists)
- User account that connected the device

2.4 Memory Analysis

Use Volatility 3 (preferred) or Volatility 2. Document every plugin/command run and its output.

SANS METHOD	Six-part memory methodology: (1) Identify rogue processes, (2) Analyse processes & DLLs, (3) Review network artefacts, (4) Look for code injection, (5) Check for rootkit signs, (6) Dump & scan suspicious processes/drivers.

- Document all plugins/commands run and link to output files
- Key areas to examine:
  - Process list: pslist, pstree, psscan (compare for hidden processes)
  - DLL analysis: dlllist, ldrmodules (check for unlinked/injected DLLs)
  - Network artefacts: netscan, netstat (connections, listening ports)
  - DNS cache: extract cached DNS entries
  - Command history: cmdline, consoles, cmdscan
  - Code injection: malfind, hollowfind (detect injected/hollow processes)
  - Rootkit indicators: ssdt, driverirp, callbacks, idt
  - Loaded devices and drivers: driverscan, modscan
  - Suspicious driver dump: moddump, procdump for manual inspection
- Did output match expectations? If not — document discrepancy and explain in report
- Volatility version used (v2 vs v3 — plugin names differ significantly)

2.5 Master File Table (MFT) & File System Artefacts

- MFT analysis: file creation, modification, access, and MFT entry change ($STANDARD_INFORMATION vs $FILE_NAME)
- Look for timestomping: $SI vs $FN timestamp discrepancies are a key indicator
- Deleted file recovery: carve from unallocated space (Autopsy, PhotoRec, Scalpel)
- Alternate Data Streams (ADS): Zone.Identifier shows file origin (internet download)
- $I30 slack: directory index slack can contain deleted file entries
- $UsnJrnl / $LogFile: change journal for file activity history
- Prefetch files: evidence of execution (C:\Windows\Prefetch*.pf) — document with PECmd
- Amcache.hve / Shimcache: program execution history even after deletion
- LNK files: recent files, shortcuts, auto-created on first access (LECmd)
- Jump lists: recent documents and application usage (JLECmd)
- Shellbags: folder browsing history, even for deleted/external locations (SBECmd)

2.6 Network Cache & Connections

- DNS cache: ipconfig /displaydns output or from memory dump
- Hosts file: check for malicious entries in C:\Windows\System32\drivers\etc\hosts
- Network connections: active and recently terminated (netstat -naob, from memory)
- ARP cache: for lateral movement detection
- Firewall logs: inbound/outbound blocked/allowed connections
- VPN / proxy artefacts: configured endpoints, authentication logs
- RDP artefacts: Terminal Services logs, RDP bitmap cache, Recent Servers registry key

2.7 Browser Data

- Search history — document keyword searches and timestamps
- Cache analysis — cached web pages and resources
- Known bad web domains / IPs visited
- Download history — files downloaded, origin URL, timestamp
- Cookies and session tokens
- Browser extensions — any suspicious or unknown extensions installed?
- Browser databases: Chrome/Edge (SQLite), Firefox (places.sqlite) — use Hindsight, BrowsingHistoryView
- Private / incognito mode evidence (limited, but some artefacts remain in memory)

2.8 Email Analysis

- Phishing / spear-phishing attempts — identify sender, subject, delivery infrastructure
- Business Email Compromise (BEC) indicators — financial fraud, wire transfer requests
- Attempts to exfiltrate information via email
- Attachments — hash all attachments, submit to sandbox if approved
- Keyword search across email corpus — document keyword list, count, and rationale
- Email headers: Received chain, SPF/DKIM/DMARC results, X-Originating-IP
- PST / OST / MBOX / EML file formats — document tool used (Autopsy, MailXaminer, Kernel OST Viewer)
- Deleted email recovery from unallocated space or mail store

2.9 Webshells & Server-Side Artefacts

- Is the system exposed to the internet? Document exposure and any web services running
- Rationale for webshell search — what led to this hypothesis?
- Common webshell indicators: .php, .aspx, .jsp files in web root with eval(), base64_decode(), system()
- Tools: LOKI, Yara rules, manual grep, Sigma rules for log correlation
- IIS / Apache / Nginx access logs: look for unusual POSTs, 404 clusters, user-agent anomalies
- Access log timeline correlation with other system events

2.10 Pictures, Videos & Media Files

- EXIF metadata: GPS location, device make/model, timestamps, software
- File size and codec analysis: identify manipulation or re-encoding

! ILLEGAL CONTENT	For cases involving illegal images/media: ensure law enforcement is involved BEFORE proceeding. DO NOT carry out the investigation without police involvement. Follow your organisation's CSAM protocol strictly.

- Hash verification against known-bad databases (PhotoDNA, NCMEC hash sets) — law enforcement only
- Steganography analysis if suspicion exists — tools: Stegdetect, zsteg, binwalk
- Calculate SHA-256 hashes for all media files of interest
- Document search constraints: file size filters, date ranges, file type filters used to narrow scope

2.11 Keyword Search

- Document the full word list used
- How many terms were searched?
- Rationale for each term — why is it relevant to this case?
- Search scope: which data sources were included in the search?
- Results: hit count per term, locations, and context snippets
- Exclusion list: what was deliberately excluded and why?

2.12 Timeline Analysis

- Construct a super-timeline from all available sources
- Tools: Plaso / log2timeline (preferred), Autopsy timeline, Timesketch for visualisation
- Event types to include: file system (MACB), registry, event logs, browser, prefetch, LNK, shellbags
- Key activities to identify:
  - File copy, move, creation, deletion, access
  - User logon/logoff events
  - Application execution times
  - USB connection/disconnection
  - Network connection establishment
  - Timestomping detection: compare $STANDARD_INFORMATION vs $FILE_NAME timestamps
- Anchor events to known good timestamps to validate the timeline
- Document timezone normalisation applied to all sources

2.13 Filter & Targeted Searches

- Does the analysis tool support targeted ingest modules?
  - Autopsy: document which ingest modules were enabled and their configuration
  - Magnet AXIOM / EnCase: document applied artefact categories and filter sets
  - X-Ways: document filter profiles and column layouts used

DUAL-TOOL	CRITICAL: Key findings should be verified with a second independent tool wherever possible. Document both results. Discrepancies between tools must be investigated and explained in the report.

2.14 Registry Analysis

Windows registry hives are a primary source of forensic artefacts. Document all hives analysed and the tool used (RegRipper, Registry Explorer / RECmd by Eric Zimmerman).

- System hives (C:\Windows\System32\config\):
  - SAM: local user accounts, password hashes, last logon
  - SECURITY: LSA secrets, cached domain credentials
  - SOFTWARE: installed software, OS configuration, run keys
  - SYSTEM: hardware, services, timezone, network interfaces, USB history
- User hives (C:\Users\<username>\NTUSER.DAT and UsrClass.dat):
  - HKCU\Software\Microsoft\Windows\CurrentVersion\Run (autorun persistence)
  - UserAssist: encoded execution history (ROT13)
  - RecentDocs, OpenSavePidlMRU: recently accessed files
  - MuiCache, AppCompatCache (Shimcache): program execution
  - Shellbags (UsrClass.dat): folder browsing history
  - TypedPaths: addresses typed in Explorer address bar
  - Network artefacts: NetworkList (known Wi-Fi networks + first/last connection)
- Persistence locations to check:
  - HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run and RunOnce
  - HKLM\SYSTEM\CurrentControlSet\Services (service installation)
  - HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options
  - AppInit_DLLs, Winlogon notification packages, COM hijacking keys

2.15 Shell & Command History

- Windows: PowerShell ConsoleHost_history.txt, PSReadLine logs, ScriptBlock Event ID 4104
- Windows: cmd.exe history (limited — not persistent by default, check memory)
- Linux / macOS: ~/.bash_history, ~/.zsh_history, ~/.sh_history
- Key commands to look for: curl, wget, chmod, sudo, cp, mv, grep, ps, tar, ftp, tftp, nc (netcat), base64
- Python / Ruby one-liners used for download or execution
- Timestamped history if available — correlate with timeline
- Root / sudo command history: /var/log/auth.log, /var/log/secure, sudoers changes

2.16 Event Log Analysis

Windows Event Logs are critical forensic sources. Use EvtxECmd (Zimmerman) + Timeline Explorer or Chainsaw for rapid triage.

- Security log (System32\winevt\Logs\Security.evtx):
  - 4624 / 4625: Logon success / failure
  - 4648: Logon with explicit credentials (pass-the-hash indicator)
  - 4720 / 4722 / 4728: Account created, enabled, added to group
  - 4732 / 4756: Member added to local/domain Administrators
  - 4688 / 4689: Process creation / termination (with command line if auditing enabled)
  - 4698 / 4702: Scheduled task created / modified
  - 4663 / 4656: Object access (files, registry)
  - 4771 / 4768: Kerberos pre-auth failed / ticket requested
- System log: Service installation (7045), driver loading, system startup/shutdown
- Application log: application crashes, software installation
- PowerShell logs: Event IDs 4103, 4104 (script block), 4105, 4106
- Sysmon (if deployed): Event ID 1 (process create with hash), 3 (network), 11 (file create), 13 (registry)
- RDP: Microsoft-Windows-TerminalServices-LocalSessionManager/Operational (21, 22, 23, 25)
- Cleared logs: Event ID 1102 (Security log cleared), 104 (System log cleared) — major indicator

3 — Report of Output

GOVERNANCE Report structure must be defined BEFORE the incident — agreed and accepted by C-level, IT Governance, and the IT Security team. Do not improvise report format during a live incident.

3.1 Document Overview

- Reference to ticket / incident / event (case ID)
- Quick summary of case type and status
- Document classification level
- Version history and author(s)

3.2 Executive Summary

Non-technical. Maximum 1–2 pages. Designed for senior management and legal counsel.

- Summary of events in plain language
- Graphic timeline of key events (visual)
- Root cause of the incident
- Business impact assessment
- High-level recommendations

3.3 Abbreviations & Definitions

- List all technical abbreviations used in the report
- Define all forensic terms that may be unfamiliar to non-technical readers
- Reference to applicable legal frameworks or standards (e.g. GDPR, ISO 27037)

3.4 Incident Investigation Report

- Driver for the forensic investigation — what triggered it?
- Crime / policy violation details — what was the illegal intent or AUP violation?
- Network overview: diagram of affected infrastructure
- Forensics analysis overview: scope, methods, and tools
- Containment actions taken: isolation, account suspension, blocking
- Findings, root cause, and key artefacts — referenced by hash and exhibit number
- Remediation steps taken
- Recommendations for prevention

3.5 Forensic Technical Report

- Examiner background and qualifications
- Tools used — name, version, and any known issues or limitations
- Tool output — referenced and stored as evidence
- Dual-tool verification results where applicable
- Collection method and chain of custody documentation
- Hash verification of all evidence items (before and after analysis)
- All findings reproducible: another qualified examiner must reach the same conclusion

3.6 Addenda

- Appendix A: Full IOC list (machine-readable: CSV, JSON, STIX/TAXII)
- Appendix B: Tool output logs
- Appendix C: Chain of custody forms
- Appendix D: Hash manifest for all evidence items
- Appendix E: MITRE ATT&CK mapping table (if applicable)
- Appendix F: Raw data exports (keyword hit lists, timeline exports, etc.)

A forensic analysis is only as strong as its documentation. Every hypothesis tested, every tool run, every artefact found — must be recorded with precision, objectivity, and reproducibility.